APPLICATION UNDER UNITED STATES PATENT LAWS

Atty. Dkt. No. 014271/0259668
(M#)

Invention: *"Method and System to Provide Modular Parallel Precoding in Optical Duobinary Transmission Systems"*

Inventor (s): Ricky Keang-Po **HO** and Joseph M. **KAHN**

Pillsbury Winthrop LLP
Intellectual Property Group
1100 New York Avenue, NW
Ninth Floor
Washington, DC 20005-3918
David A. Jakopin
Telephone: **650/233-4790**

09847966.040401

This is a:

☐ Provisional Application

☒ Regular Utility Application

☐ Continuing Application
☐ The contents of the parent are incorporated by reference

☐ PCT National Phase Application

☐ Design Application

☐ Reissue Application

☐ Plant Application

☐ Substitute Specification
Sub. Spec Filed __________
in App. No. ___/________

☐ Marked up Specification re
Sub. Spec. filed __________
In App. No ___/________

SPECIFICATION

APPLICATION FOR UNITED STATES PATENT

INVENTORS

Keangpo Ho; Joseph M. Kahn

METHOD AND SYSTEM TO PROVIDE MODULAR PARALLEL PRECODING IN OPTICAL DUOBINARY TRANSMISSION SYSTEMS

FIELD OF THE INVENTION

The present invention is directed to communications systems, and more particularly to systems and methods for calculating the cumulative parity of a binary number sequence using modular based parallel processing.

BACKGROUND

It is well known that in optical communication systems conveying digital information, whether the digital information is transmitted as single signal at a single carrier wavelength or as multiple signals at different carrier wavelengths (i.e., wavelength-division multiplexing), for a fixed bit rate per carrier wavelength, it is beneficial to design the transmitted signal to have a narrow optical spectrum. The narrow optical spectrum allows two wavelength-divisionmultiplexed channels close to each other, and usually provides more tolerance to the chromatic dispersion of the optical fiber.

Numerous patents and research papers have documented the use of on-off keying with duobinary filtering in optical communication systems. All of these works have utilized precoding to permit symbol-by-symbol detection without error propagation. While those works have described many different techniques to implement precoding, duobinary filtering, and modulation of the duobinary signal onto the optical carrier, all of these techniques result in transmission of equivalent optical signals, which take on one of three possible electric-field amplitude values, e.g., $\{-a, 0, a\}$. With a precoder, it is possible to recover the transmitted information bits by performing symbol-by-symbol detection on a signal proportional to the received optical intensity, such as the photocurrent in a direct-detection receiver. This technique also narrows the optical spectrum by about a factor of two as compared to on-off keying.

FIG. 1 is a block diagram illustrating a precoder **1** and a duobinary filter **2** as implemented in a transmitter in a conventional optical duobinary transmission system. To facilitate symbol-by-symbol detection, as shown in FIG. 1, the precoder **1** is used before the duobinary filter **2**. Between the precoder **1** and the duobinary filter **2**, the level shifter (L/S) **3** changes a logic value of "1" to a positive amplitude value of $a/2$ and a logic value of "0" to a negative amplitude value of $-a/2$. The precoder **1** is formed by an exclusive-OR (XOR) gate circuit **10** and a one-bit delay **7**. The precoder **1** inverts the logical value of the output **5** only when the logical value of its input signal **4** is "1", and maintains the logical value of the output when the logical value of its input signal is "0". The logical value of the output **5**, delayed by the one-bit delay **7** is fed back to an input of the XOR gate **10**. Mathematically, the precoder **1** calculates the cumulative parity of the binary number input sequence **4**.

The duobinary filter **2** separates the signal to two branches, one of the branches is delayed by a one-bit delay **8** and combined with another branch without delay at a summer **9**. The output **6** of the duobinary filter **2** is usually loss-passed and sent to an external modulator in particular, and an optical modulation subsystem in general.

In the precoder **1** of FIG. 1, the precoding circuit has to operate in the same rate as the serial binary input **4**. Problems generally occur for high data transmission rates, for example, 10-, 40-, 80-, 100-, and 160-Gb/s input signals. First, a high-speed XOR gate may not be available or may be quite expensive. Second, the realization of one-bit delay for the XOR gate is difficult. The one-bit delay **7** can utilize the propagation time of the feedback transmission line or can use a D-type flip-flop. If the propagation delay of the XOR gate **10** cannot be ignored compared with a time-slot of one bit due to the increase of the transmission rate, the delay time for the feedback to the XOR gate would become longer than one time-slot time.

Referring to FIG. 2, it is a block diagram illustrating the detailed configuration **20** of a conventional differential precoder as described in the prior art. For example, parallel precoding circuits are described in the European patent application of EP 1 026 863 A2 filed March 2, 2000 and published September 18, 2000, the paper of Yoneyama et al. ("Differential Precoder IC Modules for 20- ad 40-Gbit/s Optical Duobinary Transmission Systems," IEEE Transactions on Microwave Theory and Techniques, vol. 47, no. 12, Nov. 1999, pp. 2263-2270), and the paper of K. Murata et al. ("Parallel precoder IC module for 40-Gbit/s optical duobinary transmission systems," Electronics Letters, vol. 36, no. 18, Aug. 31, 2000, pp. 1571-1572). The circuit **20** of

FIG. 2 uses a multiple input XOR gate **31** to calculate the parity of *K* sets of parallel data **30**, followed by a differential circuit **33** similar to the precoder **1**, a one-bit delay **37**, and a ladder of XOR gates **32** to calculate each of the individual outputs **40**. The multi-input XOR gate **31** is by itself a very complicated circuit, requiring many two-input logic gates. One implementation of the multi-input XOR gate can use a ladder of XOR gates. Another implementation of the multi-input XOR gate uses a tree of XOR gates. As shown in the papers of Yoneyama et al. and Murata et al., the circuit **20** requires elaborate circuit elements to align the timing of all *K* output data. For simplicity, the circuit elements for timing alignment are not shown in FIG. 2. In FIG. 2, the output of **40(K)** has no gate delay but the output of **40(1)** has (K-1) gate delays from the XOR gates of **10(K-1)** to **10(1)** in the ladder of XOR gates **32**. As an indication of the difficulty, a four-input circuit in Yoneyama et al. requires two separate integrated circuits (ICs) occupied mostly by many electrical components used to compensate for gate delay. The requirement of timing alignment makes the prior parallel precoding circuits of EP1,026,863, Yoneyama et al., and Murata et al. for the parallel precoder very difficult to implement, especially for very large number of parallel inputs *K*.

Needed is a precoder design that can manage timing issues while accommodating large numbers of parallel inputs efficiently.

SUMMARY

According to one aspect of the present invention, a circuit using modular based parallel processing calculates the cumulative parity of a binary number input sequence. The circuit is used, for example, to implement a precoder for an optical duobinary transmission system. The design permits a relatively low-speed circuit to be used as the precoder before a time-division multiplexer. The parallel circuit can be scalable to process a very large number of sets of parallel binary data by the usage of two basic modules, namely, a parity module and a delay module.

A circuit to calculate the cumulative parity of a binary number sequence according to a presently preferred embodiment is presented in another aspect of the present invention. The circuit includes an array of functional modules. The modules are aligned to form columns and rows within the array. The array is configured to receive the binary number sequence at a first column of the modules. The array is configured to produce the cumulative parity as output at a last column of the modules. Each module is either a parity module or a delay module. A parity module is configured to receive certain input bits from either the binary number sequence or from a previous column and to calculate the parity of the certain input bits. A delay module is configured to receive other input bits from either the binary number sequence or from a previous column and to delay the other input bits.

A circuit to calculate the cumulative parity of a binary number sequence according to a presently preferred embodiment is presented in another aspect of the present invention. The circuit includes an array of delay elements, diagonal gate elements, and column gate elements. The delay elements are aligned to form M + 1 columns and M rows within the array, where M represents a number of parallel input bit values. The array is configured to receive the binary number sequence at the first column of the delay elements and to produce the cumulative parity as output at the (M+1)th column of the delay elements. The array includes diagonal delay elements, non-diagonal delay elements, and (M+1)th column delay elements. The diagonal delay elements form a diagonal of an M column by M row inner array of the array, from the first row and the first column to the Mth row and the Mth column of the array. The non-diagonal delay elements are the remaining delay elements within the inner array. The diagonal gate elements are located from the second row through the Mth rows of the array. The diagonal gate elements calculate parity information. The diagonal gate elements each have a diagonal gate output connected to a diagonal delay input of the corresponding diagonal delay element in the same row and the next column of the array, a first diagonal gate input connected to a diagonal delay output of the corresponding diagonal delay element in the prior row and the previous column of the array, and a second diagonal gate input connected to a non-diagonal delay output of the corresponding non-diagonal delay element in the same row and the previous column of the array. The column gate elements are located from the first row to the Mth row of the array and between the Mth column and the (M+1)th column of the array. The column gate elements each having a column gate output connected to a column delay input of the corresponding (M+1)th column delay element in the same row of the array. The column gate elements are used to pass the parity information from the diagonal and non-diagonal outputs of respective diagonal and non-diagonal delay elements in prior columns of the array to the (M+1)th column delay elements.

A method of using an array of M(M+1) modules to calculate the cumulative parity of a binary number sequence according to a presently preferred embodiment is presented in another aspect of the present invention. The array includes M rows of M+1 modules and M+1 columns of M modules. Within a first clock cycle T, the cumulative parity of a first input group of n input bit values and a first initial parity input value is calculated at the first row first column module, a second input group of n input bit values is delayed at the second row first column module, and an

Mth input group of n input bit values is delayed at the Mth row first column module. Within a second clock cycle 2T, the cumulative parity of the first input group is delayed at the first row second column module, the cumulative parity of the second input group and a second initial parity input bit value is calculated at the second row second column module, and the Mth input group is delayed at the Mth row second column module. Within an Mth clock cycle MT, the cumulative parity of the first input group is delayed at the first row Mth column module, the cumulative parity of the second input group is delayed at the second row Mth column module, and the cumulative parity of the Mth input group and an Mth initial parity input bit value is calculated at the Mth row Mth column module. Within an (M+1)th clock cycle (M+1)T, a first output group of n output bit values is calculated at the first row (M+1)th column module, a second output group of n output bit values is calculated at the second row (M+1)th column module, and an Mth output group of n output bit values is calculated at the Mth row (M+1)th column module.

A method of calculating the cumulative parity of a binary number sequence using an array of parity and delay modules to calculate the cumulative parity of a binary number sequence according to a presently preferred embodiment is presented in another aspect of the present invention. The array includes M rows of M+1 modules and M+1 columns of M modules. The binary number sequence is received at a series of inputs at the first column of the array. Parity information is calculated using parity modules of the array. The parity information is passed through the array, column by column, from the first column to the (M+1)th column. The timing of the parity information is aligned using delay modules of the array. The cumulative parity of the binary number sequence is provided at a series of outputs at the (M+1)th column of the array.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other features, aspects, and advantages will become more apparent from the following detailed description when read in conjunction with the following drawings, wherein:

FIG. 1 is a block diagram illustrating a precoder and duobinary filter as implemented in a transmitter in a conventional optical duobinary transmission system;

FIG. 2 is a block diagram illustrating the detailed configuration of a conventional differential precoder as implemented in a parallel circuit;

FIG. 3 is a block diagram illustrating an exemplary modular and scalable parallel precoding circuit according to a presently preferred embodiment;

FIG. 4 is a block diagram illustrating one exemplary configuration of the parity module according to FIG. 3;

FIG. 5 is a block diagram illustrating another exemplary configuration of the parity module according to FIG. 3;

FIG. 6 is a block diagram illustrating one exemplary configuration of the delay module according to FIG. 3; and

FIG. 7 is a block diagram illustrating one exemplary, four-input configuration of the precoding circuit of FIG. 3.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

According to one aspect of the present invention, a method is provided to design a precoding circuit for the generation of very high speed signals to be utilized in an optical fiber communication system in a systematic and modular way. Mathematically, the precoding circuit calculates the cumulative parity of a binary number input sequence using parallel processing. When implemented as a precoder in an optical duobinary transmission system, the precoding circuit can be used to precode the binary sequence before instead of after a time-division multiplexer.

Even with a very large number of sets of parallel input data, the circuit consists of only two basic building modules: a parity module and a delay module. Dividing a serial binary data input sequence into many sets of parallel data streams, the circuit is capable to handle very high transmission rate by a simple configuration.

The parity module calculates the cumulative parity of an initial parity input and n parallel binary data inputs, and provides n parallel outputs, preferably after one clock cycle. The delay module delays the n parallel binary data inputs, preferably for one clock cycle.

Using the precoding circuit, sets of parallel data are divided into M groups of n sets of parallel data. Preferably, all parity modules and delay modules are in row and column arrangement. There are M rows of modules for each group of parallel data. Each group of parallel data are processed using $M + 1$ columns of modules. The n parallel outputs of each module are connected to the n parallel inputs of the module in the same row and the next column. The last output bits of the parity module may connect to the initial parity input of some other parity modules.

The modular and scalable circuit can be used as the parallel precoder of a duobinary transmitter placed before a time-division multiplexer. The circuit can also be used for other applications requiring the calculation of the cumulative parity of the inputs.

The present invention will now be described in detail with reference to the accompanying drawings, which are provided as illustrative examples of preferred embodiments of the present invention.

FIG. 3 is a block diagram illustrating an exemplary modular and scalable parallel precoding circuit **50** according to a presently preferred embodiment that incorporates aspects of the presently preferred methods and systems described herein. The precoding circuit **50** of FIG. 3 preferably uses two types of functional modules, a parity module **100** and a delay module **200**. The parity module **100** and a delay module **200** are preferably implemented by circuits, examples of which are described in more detail below. All of the parity modules **100** and the delay modules **200** are arranged in an array of modules having a total of *M* rows and *M* + 1 columns. *M* parity modules **100(1, 1)**, **100(2, 2)**, **…**, **100(M, M)**, that is, **100(i, i)** for *i* from 1 to *M,* are in the diagonal position of an inner array within the array of modules. The inner array has *M* rows and *M* columns, that is, the columns 1 through *M* of the array of modules. Another *M* parity modules **100(1, M+1)**, **100(2, M+1)**, **…**, **100(M, M+1)** are in the last column *M* + 1 of the array of modules. *M*(*M* – 1) delay modules **200(1, 2)**, **200 (3,1)**, **…**, **200(M-1, M)**, that is, **200(i, j)** for *i* not equal to *j*, *i* and *j* from 1 to *M,* are located in non-diagonal positions in the inner array within the array of modules.

In FIG. 3, other than the clock signal CLK **180**, each of the delay modules **200** has the same number of inputs and outputs *n*. Other than the clock signal CLK **180**, each of the parity modules **100** has $n + 1$ inputs and n outputs. The precoding circuit **50** in FIG. 3 operates with $K = Mn$ parallel sets of data as both inputs and outputs. The $K = Mn$ parallel sets of input data $D_{1,n}$, $D_{n+1,2n}$, … $D_{M(n+1)+1,Mn}$ are received at inputs of M input groups **150(1), 150(2), …, 150(M)**. Each of the input groups **150(1), 150(2), …, 150(M)** has n parallel inputs to respectively receive n sets of parallel data. The $K = Mn$ parallel sets of output data $I_{1,n}$, $I_{n+1,2n}$, … $I_{M(n+1)+1,Mn}$ are output by the circuit **50** at outputs of M output groups **160(1), 160(2), …, 160(M)**. Each of the output groups **160(1), 160(2), …, 160(M)** has n parallel inputs to respectively output n sets of parallel data.

FIGS. 4 and 5 are block diagrams illustrating two exemplary configurations of the parity module **100** according to FIG. 3. The parity module **100** has n parallel inputs **150** to respectively receive n exemplary parallel sets of input data D_1 to D_n and n parallel outputs **160** to respectively output n exemplary parallel sets of input data I_1 to I_n. The parity module **100** has an additional initial parity input of D_0 **151**. The first output of the parity module **100** is the parity of D_0 and D_1; that is, $I_1 = D_0 + D_1$ (mod 2). The second output of the parity module **100** is the parity of D_0, D_1, and D_2; that is, $I_2 = D_0 + D_1 + D_2$ (mod 2). In general, for i from 3 to n, the ith output of the parity module is the parity of D_0 to D_i; that is, $I_i = D_0 + D_1 +\ldots+ D_i$ (mod 2). In the configuration of FIG. 4, the cumulative parities of the inputs **150** and **151** are calculated by a ladder of XOR gates **140**. Another configuration to calculate the cumulative parities is shown in FIG. 5. To align the timing of the outputs **160**, a bank of D-type flip-flops **120(1), 120(2)** to **120(n)** are used, synchronized by the trigger from the clock signal CLK **180**. The last bit of the parallel output **161** may be branched out separately from the parallel outputs **160**. This special branch out, for example, the parallel output **161(1, 1),** is preferably used in parity modules **100(1, 1), 100(2, 2),**

..., **100(M-1, M-1)** located along a diagonal of the inner array and the parity module **100(M, M+1)** in the last column *M*+1 of the array in FIG. 3. In both of the exemplary configurations of FIGS. 4 and 5, the number of parallel inputs *n* is limited by the gate delays of the respective XOR ladder **140, 142** and the bank of D-type flip-flops **120(1)**, **120(2)** to **120(n)**. A conservative design goal for the delay of each gate is 1/(2*n*) of one-bit interval.

FIG. 6 is a block diagram illustrating one exemplary configuration of the delay module **200** according to FIG. 3. The delay module of FIG. 6 uses a bank of *n* D-type flip-flops **130(1)**, **130(1)** to **130(n)**, synchronized by the trigger from the clock signal CLK **180**.

In the precoding circuit of FIG. 3, for modules **100, 200** in the same row, the inputs of the delay module **200** are connected to the outputs of the module in the previous column, either the parity module **100** or the delay module **200**. The initial parity input **151(1, 1)** of the parity module in the first column and first row **100(1, 1)** is preferably connected to logic "0". The initial parity input of other parity modules **100** in other diagonal position **100(2, 2)** to **100(M, M)** are connected to the last bit output of the parity module **100** in the previous column and previous row, that is, **151(i, i)** connected to **161(i-1, i-1)** for all i from 2 to *M*, as an example, **151(2, 2)** to **161(1, 1)**. The *n* parallel inputs of parity modules **100** are connected to the *n* parallel outputs of the delay module **200** or the parity module **100** in the previous column in the same row. The first input of the *n* parallel inputs of a module **100, 200** is connected to the first output of either the parity module **100** or the delay module **200** in the previous column and the same row. The first input of the parity module **100** is the first parallel input, not the initial parity input. The second input of the *n* parallel inputs of a module **100, 200** is connected to the second output of either the parity module **100** or the parity module **200** in the previous column and the same row. It continues similarly through the inputs of a module **100**, **200** so that, for example, the *n*th or the last input of the *n* parallel inputs of a module **100**, **200** is connected to the *n*th or the last output of either the parity module **100** or the parity module **200** in the previous column and the same row.

In FIG. 3, the last output bit value I_{Mn} of the outputs $I_1, I_2, \ldots, I_{Mn}$, which corresponds to the last output **161(M, M+1)** of the *M*th row (*M*+1)th column parity module **100(M, M+1)**, is presented at the initial parity inputs **151(1, M+1), 151(2, M+1), ..., 151(M, M+1)** of all of the parity modules **100(1, M+1), 100(2, M+1), ..., 100(M, M+1)** in the last column $M + 1$ of the array of modules in the circuit **50**.

FIG. 7 is a block diagram illustrating one exemplary, four-input configuration **300** of the precoding circuit **50** of FIG. 3. The exemplary configuration **300** of FIG. 7 has n = 1 and K = M = 4 for four sets of parallel input and output data. The D-type flip-flops **320 (2,1), 320 (3,1), 320 (4, 1), 320 (1,2), 320 (3,2), 320 (4,2), 320 (1,3), 320 (2,3), 320 (4,3), 320 (1,4), 320 (2,4), 320 (3,4), 320 (1,5), 320 (2,5), 320 (3,5), 320 (4,5)** in the non-diagonal position are equivalent to the exemplary delay module **200** in FIG. 6 for a single input (*n*=1). The D-type flip-flop in the first column and first row **320(1, 1)** is the logic simplification of the exemplary parity module **100** in FIG. 4 for two inputs, including $D_0 = 0$ (*n*=1). That is, an XOR gate that has inputs of D_1 and D_0=0 equals D_1 at its output, so that the XOR gate is not needed for the parity module, and the input D_0=0 is not shown FIG. 7. Other D-type flip-flops **320(2, 2), 320(3, 3), 320(4, 4)** in the diagonal position combined with the respective corresponding diagonally located XOR gate **310(2), 310(3), 310(4)** in the same row, are the exemplary parity module **100** of FIG. 4 for single (non initial parity) input and output (n=1). The other input, that is, the initial parity input of the diagonally located XOR gate **310(2)**, **310(3)**, **310(4)** is connected to the output, that is, the last output with *n*=1, from the respective D-type flip-flop **220(1, 1)**, **320(2, 2)**, **320(3, 3)** in the previous row and column. The bank of D-type flip-flops **320 (1,5)**, **320 (2,5)**, **320 (3,5)**, **320 (4,5)** in the last column *M+1*, combined with the bank of XOR gates **315(1)**, **315(2)**, **315(3)**, **315(4)** is equivalent to a bank of parity modules **200** of FIG. 4 for two inputs, including the initial parity input receiving the bit value **161(4,5)**.

As used herein, the term delay element is intended broadly to refer to a circuit element that outputs the value of bits received at its input following a period of time, such as one or more clock cycles. For example, a delay element may be implemented as a D-type flip-flop. In a D-type flip flop having a one clock cycle delay, when the CLK input of the flip flop is changed from a logical zero to a logical one, the output of the flip flop reflects the logic level present at the input. When the CLK input falls to logic zero, or changes from one to zero, the last state of the input is trapped and held in the flip flop. The D-type flip flop may also be called the edge-trigged D-type flip-flop. The D-type flip-flop may be constructed by connecting Set Rest (SR) flip-flops or latches, some NAND gates, other logic gates, or other types of flip-flop together. Some memory devices can be used to function as the D-type flip-flop. Although in a presently preferred embodiment, the delay element includes the D-type flip flop, other devices are possible, such as other flip-flops, logic gates, or memory devices.

Although the present invention has been particularly described with reference to the preferred embodiments, it should be readily apparent to those of ordinary skill in the art that changes and modifications in the form and details may be made without departing from the spirit and scope of the invention. It is intended that the appended claims include such changes and modifications.

09841799.042401